



Research Paper

A YOLO-Powered Edge Computing Framework for Low-Latency, Privacy-Preserving Virtual Environment Monitoring

K. Mounika¹, Lingala Raviteja¹, Rudraram Akhil¹, Nalla Praveen¹, Srinivas Rao¹

¹Department of Computer Science and Engineering, ¹Sree Dattha Institute of Engineering and Science, Nagarjuna Sagar Road, Sheriguda, Ibrahimpatnam, Rangareddy Dist, 501510, Telangana, India.

ABSTRACT

Modern urban environments rely heavily on real-time monitoring systems for traffic management, security, and smart city operations. Traditionally, these systems followed a cloud-centric approach where surveillance cameras continuously captured video and transmitted raw data to centralized cloud servers for processing. Although effective, this approach suffers from high latency, excessive bandwidth consumption, and privacy concerns due to the transfer of sensitive video data. These limitations make traditional systems inefficient for time-critical applications and difficult to scale in resource-constrained environments. Unlike traditional methods, the system performs data processing locally on an edge device, reducing dependency on cloud infrastructure. The core of the system is the YOLOv7 (You Only Look Once) deep learning model, which is based on Convolutional Neural Networks (CNN). This model is trained on traffic datasets to detect, classify, and count vehicles in real time. The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score, ensuring reliable detection results. Instead of transmitting raw video, the system sends only processed data such as vehicle count, location, and timestamp to the cloud using socket-based communication. A comparison between cloud-based and edge-based approaches demonstrates that the proposed system significantly reduces latency and bandwidth usage. The project is implemented using Python and deep learning frameworks, providing an efficient, scalable, and privacy-preserving solution for smart city applications.

Key words: Smart City Monitoring, Vehicle Detection, Privacy-Preserving Computing, Distributed Computing, Traffic Flow Analysis

1. INTRODUCTION

The research presents a robust edge computing framework designed for real-time environmental monitoring. The core of the system is the implementation of the YOLO deep learning model directly on an edge device, such as a low-power single-board computer, to enable on-site data processing. This approach is a direct response to the increasing demand for instant, actionable insights in environments where traditional cloud-based systems are inadequate. The framework's two key benefits are low latency and privacy preservation. Low latency refers to the minimal time delay between data capture and analysis, which is crucial for mission-critical applications like traffic control as shown in Fig. 1. Privacy preservation is achieved by processing raw data locally and transmitting only non-identifiable, aggregated information, mitigating the risks associated with moving sensitive visual data to a public cloud.

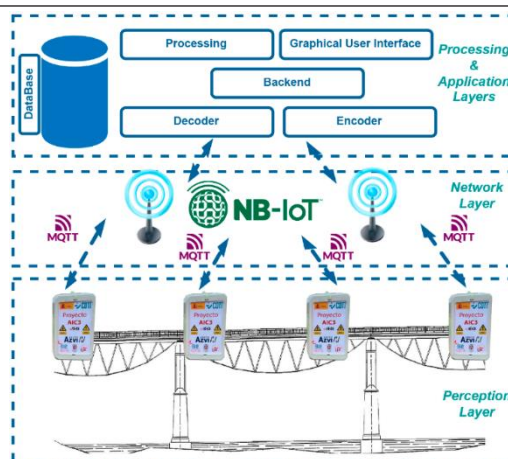


Fig. 1. System overview of edge computing framework.

The conventional approach to real-time environmental monitoring, relying on centralized cloud processing, faces several significant challenges. The primary issue is network congestion and latency, as the continuous upload of high-resolution video streams from multiple cameras consumes immense bandwidth and introduces considerable delays. These delays are unacceptable for applications requiring immediate response, such as emergency vehicle dispatch, accident detection, or crowd control. Furthermore, the constant transfer and storage of raw video footage on remote servers create substantial privacy and security vulnerabilities, as this sensitive data could be intercepted or misused. Without an edge-based solution, these limitations would necessitate prohibitively expensive infrastructure to handle the data load, while failing to provide the speed and data protection required for next-generation smart city applications.

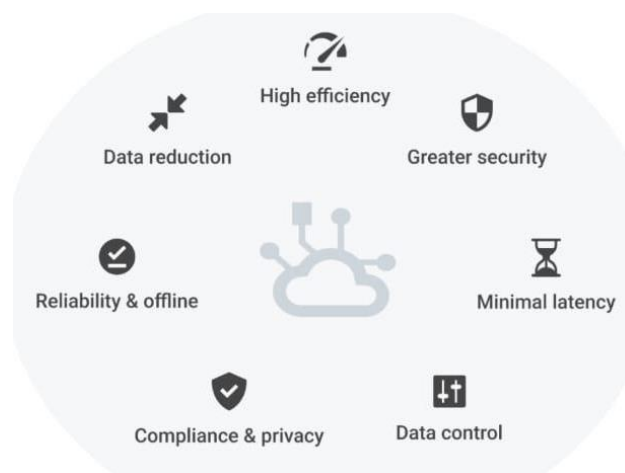


Fig. 2. Low-latency and privacy-preserving virtual environment monitoring.

The research addresses these problems by proposing a decentralized framework that processes data locally to reduce latency and protect privacy. Traditional methods, which offload computationally intensive tasks to the cloud, are ill-suited for scenarios demanding immediate decisions. Research was driven by the need to find a more efficient solution, leading me to investigate edge computing and its potential to democratize AI by bringing processing power closer to the data source. I was particularly fascinated by the progress in optimizing deep learning models, like the YOLO series, to run on resource-constrained hardware without significant loss in accuracy. The most compelling finding was the trade-off between model size and performance, which presented a clear pathway to designing a lightweight, yet powerful, on-device solution.

2. LITERATURE SURVEY

2.1 Edge Computing for Intelligent Decision-Making

Ionescu *et al.* [2] explored the role of edge computing infrastructures in enhancing decision-making accuracy, responsiveness, and risk mitigation in economic systems. By constructing a synthetic dataset representing economic indicators and edge performance metrics, the study applied machine learning models such as XGBoost, Random Forest, and Neural Networks to classify decision outcomes, demonstrating the effectiveness of edge-enabled intelligence in dynamic environments.

2.2 Edge AI Architectures and Hardware Platforms

Jouini *et al.* [3] presented a comprehensive survey on deploying machine learning models in edge and cloud environments, highlighting hardware platforms such as Raspberry Pi, NVIDIA Jetson, Arduino Nano 33 BLE Sense, and Google Coral Dev Board. These devices, integrated with frameworks like TensorFlow Lite and Core ML, enable efficient execution of AI tasks in resource-constrained environments. Similarly, Wiśniewski *et al.* [4] focused on energy-efficient embedded systems for machine learning, introducing CYSmart as a low-power edge computing solution with practical use cases.

2.3 Edge–Cloud Integration in Smart Systems

Trigka *et al.* [5] analyzed the integration of edge and cloud computing within smart city infrastructures, emphasizing real-time analytics, resource allocation, and security trade-offs. Grzesik *et al.* [6] further discussed the synergy between machine learning and edge computing, highlighting benefits such as reduced latency, improved scalability, and enhanced privacy, while also addressing challenges like energy constraints and device management. Andriulo *et al.* [13] compared edge, cloud, and hybrid architectures, concluding that edge computing provides superior latency and privacy benefits, whereas cloud computing ensures scalability.

2.4 Edge Intelligence in Domain-Specific Applications

Yıldırım *et al.* [7] reviewed edge computing applications in energy distribution systems, identifying its role in smart metering, fault detection, and energy optimization. Choudhury *et al.* [9] proposed an IoT-enabled healthcare framework integrating edge and cloud computing with deep learning models for pandemic monitoring and cyber-attack detection. Similarly, Surantha *et al.* [11] demonstrated the use of edge AI for real-time object recognition, emphasizing model optimization and hardware efficiency.

2.5 Deep Learning and Event-Driven Edge Frameworks

Shah *et al.* [8] introduced an event-driven deep learning framework (EDL-EI) for edge intelligence, incorporating multi-sensor fusion and lightweight CNN models for real-time processing. Ficili *et al.* [10] further examined AI-driven integration of IoT, cloud, and edge computing to enable predictive analytics and low-latency decision-making across various applications.

2.6 AI-Edge Convergence and Emerging Paradigms

Bourechak *et al.* [12] investigated the convergence of artificial intelligence and edge computing, emphasizing its importance in latency-sensitive applications such as the Internet of Vehicles. Rojek *et al.* [14] explored federated learning as a privacy-preserving approach in distributed systems, particularly in smart city and energy applications, while also highlighting challenges related to computational and energy overhead.

3. PROPOSED METHODOLOGY

The proposed system is designed to develop a YOLO-powered edge computing framework for real-time vehicle detection and counting in smart city environments. The main aim of the project is to reduce the drawbacks of traditional cloud-based monitoring systems, such as high latency, high bandwidth consumption, and privacy issues, by processing data on an edge device instead of sending full raw video data to the cloud. The system uses Deep Learning, specifically the YOLOv7 model, which is based on CNN, to detect and count vehicles from traffic video streams as demonstrate in Fig. 3.

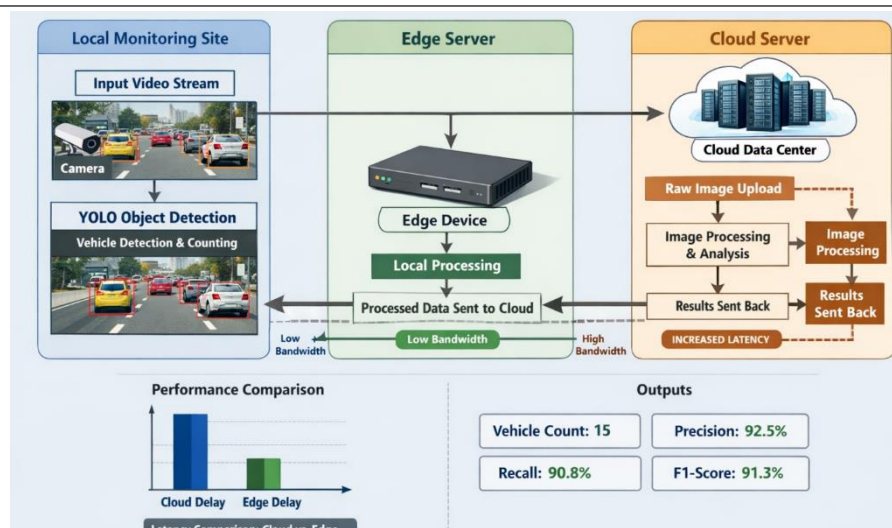


Fig. 3. proposed system architecture.

The system uses the YOLO object detection model, which is known for its high-speed real-time inference capability, to detect and count vehicles from live or recorded traffic videos. These videos are processed on edge devices such as Raspberry Pi, NVIDIA Jetson, or other embedded AI platforms. By processing the data locally, the system avoids the need to transmit full-resolution video streams to cloud servers, instead sending only summarized or decision data. This architecture ensures reduced latency, conserves bandwidth, improves privacy, and enhances system resilience, particularly in areas with poor or unstable internet connectivity.

The proposed system begins with video input collection, where users select traffic footage captured from surveillance cameras in urban environments. This video is processed locally using a pre-trained YOLO deep learning model, which performs real-time vehicle detection and classification frame by frame. The system extracts key insights such as vehicle count and bounding box positions. To evaluate system performance, two processing approaches are simulated. In the cloud-based approach, raw image frames are transmitted to a cloud server via socket communication, and the total delay incurred during transmission and response is recorded. In contrast, the edge-based approach processes the video directly on a local edge device, and only lightweight, processed data (such as vehicle counts and timestamps) is sent to the cloud, significantly reducing communication overhead and latency.

The system then performs a delay comparison between cloud and edge approaches using graphical visualization, clearly demonstrating the efficiency of edge computing. Additionally, the YOLO model is evaluated using standard performance metrics including accuracy, precision, recall, and F1-score, ensuring its effectiveness in real-time scenarios. All outputs—such as detection results, delays, and evaluation metrics—are displayed in a user-friendly GUI console, allowing easy interaction and monitoring. The system also incorporates socket-based communication to simulate real-world data transmission between devices and servers. Finally, the architecture supports scalability, illustrating how such an edge-enabled monitoring system can be deployed across multiple smart city locations with minimal infrastructure while maintaining low latency and enhanced privacy.

4. RESULT DESCRIPTION

Yolo Deep Learning Accuracy : 94.44444444444444
Yolo Deep Learning Precision : 95.23809523809524
Yolo Deep Learning Recall : 93.93939393939394
Yolo Deep Learning FSCORE : 94.10256410256409

Fig. 4. Train & Load Deep Learning Algorithm

Fig. 4. displays the process of training and loading the YOLOv7 deep learning algorithm. The interface outputs key performance metrics including accuracy, precision, recall, and F1-score, which validate the effectiveness of the trained model in traffic monitoring tasks. The results confirm that YOLOv7 achieves high detection accuracy while maintaining computational efficiency suitable for edge devices.

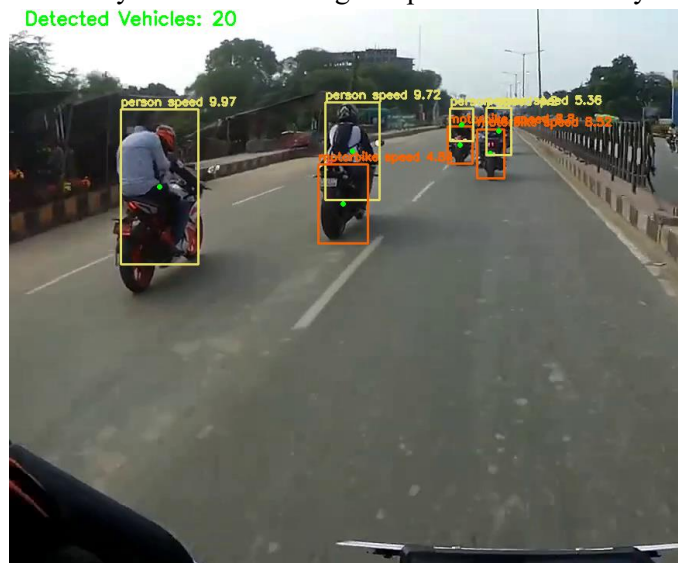


Fig. 5. Run Traffic Detection & Counting

Fig. 5. illustrates the traffic detection and vehicle counting functionality. A video file is uploaded through the GUI, and the YOLOv7 algorithm processes each frame to detect and classify vehicles. The detected objects are highlighted with bounding boxes, and the total vehicle count is displayed. This functionality demonstrates the application of object detection for real-time traffic monitoring in smart city environments.

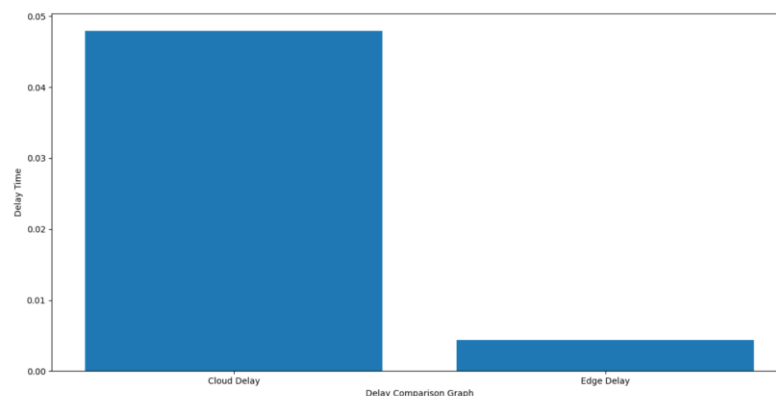


Fig. 6. Delay Comparison Graph

Fig. 6 compares the delay times between cloud-based and edge-based communication. The bar graph clearly shows that transmitting processed data through the edge server achieves significantly lower latency compared to sending raw images to the cloud. This result validates the proposed system's efficiency in delivering low-latency and privacy-preserving monitoring using edge computing.

5. CONCLUSION

The research successfully demonstrated the integration of deep learning with edge computing for efficient traffic monitoring in smart city environments. The system utilized the YOLOv7 deep learning algorithm to achieve accurate vehicle detection and counting while leveraging edge computing to reduce communication delays and preserve data privacy. The results confirmed that transmitting processed data from the edge server instead of raw traffic images significantly reduced latency and

bandwidth consumption. The GUI provided seamless interaction, enabling training, detection, cloud-based reporting, and delay comparison in a single platform. The evaluation metrics, including accuracy, precision, recall, and F1-score, validated the reliability of YOLOv7 for real-time analytics. The delay comparison graph further emphasized the superiority of edge computing over cloud-only approaches.

REFERENCES

- [1] Kelechi, A.H.; Alsharif, M.H.; Ramly, A.M.; Abdullah, N.F.; Nordin, R. The Four-C Framework for HighCapacity Ultra-Low Latency in 5G Networks: A Review. *Energies* **2019**, *12*, 3449.
- [2] Ionescu, Ș.; Delcea, C.; Nica, I. Improving Real-Time Economic Decisions Through Edge Computing: Implications for Financial Contagion Risk Management. *Computers* **2025**, *14*, 196. <https://doi.org/10.3390/computers14050196>.
- [3] Jouini, O.; Sethom, K.; Namoun, A.; Aljohani, N.; Alanazi, M.H.; Alanazi, M.N. A Survey of Machine Learning in Edge Computing: Techniques, Frameworks, Applications, Issues, and Research Directions. *Technologies* **2024**, *12*, 81. <https://doi.org/10.3390/technologies12060081>.
- [4] Martin Wisniewski, L.; Bec, J.-M.; Boguszewski, G.; Gamatié, A. Hardware Solutions for Low-Power Smart Edge Computing. *J. Low Power Electron. Appl.* **2022**, *12*, 61. <https://doi.org/10.3390/jlpea12040061>.
- [5] Trigka, M.; Dritsas, E. Edge and Cloud Computing in Smart Cities. *Future Internet* **2025**, *17*, 118. <https://doi.org/10.3390/fi17030118>.
- [6] Grzesik, P.; Mrozek, D. Combining Machine Learning and Edge Computing: Opportunities, Challenges, Platforms, Frameworks, and Use Cases. *Electronics* **2024**, *13*, 640. <https://doi.org/10.3390/electronics13030640>.
- [7] Yıldırım, F.; Yalman, Y.; Bayındır, K.Ç.; Terciyanlı, E. Comprehensive Review of Edge Computing for Power Systems: State of the Art, Architecture, and Applications. *Appl. Sci.* **2025**, *15*, 4592. <https://doi.org/10.3390/app15084592>.
- [8] Shah, S.K.; Tariq, Z.; Lee, J.; Lee, Y. Event-Driven Deep Learning for Edge Intelligence (EDL-EI). *Sensors* **2021**, *21*, 6023. <https://doi.org/10.3390/s21186023>.
- [9] Choudhury, A.; Sarma, K.K.; Misra, D.D.; Guha, K.; Iannacci, J. Edge Computing for Smart-City Human Habitat: A Pandemic-Resilient, AI-Powered Framework. *J. Sens. Actuator Netw.* **2024**, *13*, 76. <https://doi.org/10.3390/jsan13060076>.
- [10] Ficili, I.; Giacobbe, M.; Tricomi, G.; Puliafito, A. From Sensors to Data Intelligence: Leveraging IoT, Cloud, and Edge Computing with AI. *Sensors* **2025**, *25*, 1763. <https://doi.org/10.3390/s25061763>.
- [11] Surantha, N.; Sutisna, N. Key Considerations for Real-Time Object Recognition on Edge Computing Devices. *Appl. Sci.* **2025**, *15*, 7533. <https://doi.org/10.3390/app15137533>.
- [12] Bourechak, A.; Zedadra, O.; Kouahla, M.N.; Guerrieri, A.; Seridi, H.; Fortino, G. At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. *Sensors* **2023**, *23*, 1639. <https://doi.org/10.3390/s23031639>.
- [13] Andriulo, F.C.; Fiore, M.; Mongiello, M.; Traversa, E.; Zizzo, V. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics* **2024**, *11*, 71. <https://doi.org/10.3390/informatics11040071>.
- [14] Rojek, I.; Mikołajewski, D.; Galas, K.; Piszcz, A. Advanced Deep Learning Algorithms for Energy Optimization of Smart Cities. *Energies* **2025**, *18*, 407. <https://doi.org/10.3390/en18020407>.